Theoretical Computer Science 48 (1986) 53-73
North-Holland

PROPERTIES OF COMPLEXITY MEASURES FOR PRAMs AND WRAMs

Siegfried BUBLITZ*, Ute SCHÜRFELD and Ingo WEGENER*
Fachbereich Informatik, Johann Wolfgang Goethe-Universität, 6000 Frankfurt am Main, Fed. Rep. Germany

Bernd VOIGT
Fakultät für Mathematik, Universität Bielefeld, 4800 Bielefeld, Fed. Rep. Germany

Communicated by M.S. Paterson
Received January 1986
Revised June 1986

Abstract. The computation of Boolean functions by parallel computers with shared memory (PRAMs and WRAMs) is considered. In particular, complexity measures for parallel computers like critical and sensitive complexity are compared with other complexity measures for Boolean functions like branching program depth and length of prime implicants and clauses.

The relations between these complexity measures and their asymptotic behaviour are investigated for the classes of Boolean functions, monotone functions and symmetric functions.

1. Introduction

A parallel random access machine is a set of processors communicating only via a shared memory. We may allow the computation power of a single processor to be unlimited. Note that the computation of a Boolean function does not become trivial then, as each processor may read at most one cell of the common memory during a computation step and may try to write in at most one of these cells.

Different processors may read from the same cell simultaneously. For PRAMs, by definition, a program is only valid if it never happens that two distinct processors try to write into the same cell at the same time. For WRAMs, different processors may try to write into the same cell simultaneously. However, the conflict is solved in such a way that only the processor with the largest number succeeds and all others fail to write.

The common memory may be restricted to m memory cells. The corresponding models are denoted by PRAM(m) and WRAM(m).

* Supported in part by DFG Grant We 1066/1-1.

For more detailed explanations of these models see, e.g., [1, 10].

For our purposes it suffices to know that lower bounds for the time complexities of PRAMs and WRAMs can be expressed in terms of the *sensitive complexity*, respectively the *critical complexity*. Moreover, for parallel computers with a realistic computation power of the single processors, these lower bounds are often even tight.

Next we define the sensitive and critical complexity and cite the known lower bounds for the time complexity of PRAMs and WRAMs.

Definition 1.1. For a Boolean function on n variables $f \in B_n$, an input vector $\boldsymbol{a}$ is called *k-sensitive* if for every $(k-1)$-element set $J \subseteq \{1, \ldots, n\}$ there exists an input vector $\boldsymbol{b}$ such that $b_j = a_j$ for $j \in J$ but $f(\boldsymbol{b}) \neq f(\boldsymbol{a})$. Thus f must not be constant on any $(n-k+1)$-dimensional subcube of $\{0,1\}^n$ containing $\boldsymbol{a}$. The *sensitive complexity* $s(f, \boldsymbol{a})$ of f at $\boldsymbol{a}$ is the maximal k such that f is k-sensitive at $\boldsymbol{a}$. Furthermore,

$$s_{\max}(f) = \max\{s(f, \boldsymbol{a}) \mid \boldsymbol{a} \in \{0,1\}^n\} \quad \text{and} \quad s_{\min}(f) = \min\{s(f, \boldsymbol{a}) \mid \boldsymbol{a} \in \{0,1\}^n\}.$$

Theorem 1.2 (Vishkin and Wigderson [10]). *The time complexity of a* PRAM(m) *computing f is* $\Omega((s_{\max}(f)/m)^{1/3})$ *while the time complexity of a* WRAM(m) *computing f is* $\Omega((s_{\min}(f)/m)^{1/2})$.

Definition 1.3. The *(Hamming-)distance* $d(\boldsymbol{a}, \boldsymbol{b})$ *of vectors* $\boldsymbol{a}, \boldsymbol{b} \in \{0,1\}^n$ equals the number of j's where $a_j \neq b_j$. The *neighbourhood* $\Gamma(\boldsymbol{a})$ of $\boldsymbol{a}$ is the set of those n vectors $\boldsymbol{b}$ such that $d(\boldsymbol{a}, \boldsymbol{b}) = 1$. The *ball* $B(\boldsymbol{a}) = \Gamma(\boldsymbol{a}) \cup \{\boldsymbol{a}\}$ consists of $\boldsymbol{a}$ and $\Gamma(\boldsymbol{a})$.

Definition 1.4. The *critical complexity* $c(f, \boldsymbol{a})$ of f at $\boldsymbol{a}$ is the number of neighbours $\boldsymbol{b}$ of $\boldsymbol{a}$ such that $f(\boldsymbol{b}) \neq f(\boldsymbol{a})$.

$$c_{\max}(f) = \max\{c(f, \boldsymbol{a}) \mid \boldsymbol{a} \in \{0,1\}^n\} \quad \text{and} \quad c_{\min}(f) = \min\{c(f, \boldsymbol{a}) \mid \boldsymbol{a} \in \{0,1\}^n\}.$$

Theorem 1.5 (Cook, Dwork and Reischuk [1]). *The time complexity of a* PRAM *computing f is at least* $\log_b c_{\max}(f)$ *where* $b = (5+\sqrt{21})/2 \approx 4.79$.

By these bounds (which are tight for many important functions) the investigation of $s_{\max}$, $s_{\min}$, and $c_{\max}$ is motivated. We have included $c_{\min}$ only for the sake of completeness and will mention results on $c_{\min}$ only briefly. The use of the lower bounds is limited since these complexity measures in general cannot be computed efficiently. If f is given by clauses, their computation is NP-hard. Thus it is interesting to obtain results on the structure of these measures.

Before discussing our results we introduce three more complexity measures for Boolean functions.

The nonparallel computation model corresponding to PRAMs and WRAMs is a decision tree. A decision tree may be understood as a single processor which can read only a single input bit during one computation step but has unlimited computation power. The time complexity of decision trees is equal to the minimum depth

of branching programs, denoted by BPD(f) (for details of this model see, e.g., [7, 11]).

It will turn out that the complexity measures motivated above are related to the length of the prime implicants and prime clauses of the considered Boolean function.

Definition 1.6. A *monom* is a conjunction of literals, i.e., variables and negated variables. Its length equals the number of its literals. A monom m is a *prime implicant* of f if it is an implicant ($m(x)=1$ implies $f(x)=1$) and no proper shortening is also an implicant. The dual concept (conjunction $\leftrightarrow$ disjunction, $1 \leftrightarrow 0$) leads to *prime clauses.*

$$l_{\max}(f) = \max\{k \mid f \text{ has a prime implicant or a prime clause of length } k\},$$

$$l_{\min}(f) = \min\{k \mid f \text{ has a prime implicant or a prime clause of length } k\}.$$

The consideration of $l_{\max}(f)$ and $l_{\min}(f)$ is motivated by tight connections to the other complexity measures. Especially for the critical complexity of monotone Boolean functions, many results rely on the observation [12] that for monotone functions $l_{\max}(f) = c_{\max}(f)$.

Best possible general lower bounds on the critical complexity of all Boolean functions, all monotone functions, all symmetric functions, and all (monotone) graph properties have been proved in [8, 9, 12]. The lower bounds are the best possible in the sense that they coincide with the complexity of the easiest function in the corresponding class. Furthermore, one gets lower bounds for all functions in this class. But usually easy functions are exceptions and almost all functions are much more difficult. The notion "almost all functions of a class $C_n \subseteq B_n$ have property P" stands for the assertion

$$\lim_{n\to\infty} \frac{\#\{f \in C_n \mid f \text{ has property } P\}}{\# C_n} = 1.$$

We consider the class B_n of *all Boolean functions* on n variables, the class M_n of *monotone functions* in B_n and the class S_n of *symmetric functions* in B_n. For these classes and the introduced seven complexity measures we determine the typical complexity, i.e., the complexity of almost all functions.

Since the fact that $c_{\max}(f) = l_{\max}(f)$ for all $f \in M_n$ turned out to be so useful we look for more relations of this type. For each pair of complexity measures (c_1, c_2) of our list and each of our classes C_n of functions, we shall prove which of the following properties is fulfilled.

- $c_1 = c_2$, i.e., $c_1(f) = c_2(f)$ for all $f \in C_n$.
- $c_1 < c_2$, i.e., $c_1(f) \leqslant c_2(f)$ for all $f \in C_n$ and $c_1(f) < c_2(f)$ for some $f \in C_n$.
- $c_1 \lesssim c_2$, i.e., $c_1(f) \leqslant c_2(f)$ for almost all $f \in C_n$, but not $c_1(f) = c_2(f)$ for almost all $f \in C_n$, and $c_1(f) > c_2(f)$ for some $f \in C_n$.

Let us finally comment on the choice of the classes of functions. Obviously, one should consider the class B_n of all functions.

M_n is an important subclass with a nice structure. Knowing $l_{\min}$ and $l_{\max}$ we also know $s_{\min}$, $s_{\max}$, and $c_{\max}$. Thus the lower bounds of Theorems 1.2 and 1.5 depend, for monotone functions, only on the lengths of the shortest and longest prime implicants and prime clauses of the considered function. Finally, S_n contains many important functions (e.g., threshold functions and counting functions). Symmetric functions $f \in S_n$ may be represented by their value vectors $v(f) = (v_0, \ldots, v_n)$ such that v_i is the value of f on all inputs with exactly i ones. Given this representation we can compute the complexity of f for all complexity measures in time $O(n)$.

2. All Boolean functions

In the first part of this section we are going to study the relations between the different complexity measures while in the second part we shall describe the complexity of almost all Boolean functions with respect to all complexity measures.

Theorem 2.1. *For the class B_n of all Boolean functions the following holds:*

(a) $$0 < c_{\min} < \begin{cases} s_{\min} = l_{\min} \\ c_{\max} \end{cases} < s_{\max} < \begin{cases} l_{\max} \\ \text{BPD} \end{cases} < n.$$

(b) $s_{\min} = l_{\min} \leqslant c_{\max}$.

(c) $l_{\max} \leqslant \text{BPD}$.

The meaning of Fig. 1 is the following. Two complexity measures c_1 and c_2 are connected by an edge and c_2 lies above c_1 if $c_1 < c_2$. A dotted edge means $c_1 \leqslant c_2$.

Proof of Theorem 2.1. The following relations in Theorem 2.1 are obvious.

$$0 \leqslant c_{\min}(f) \leqslant c_{\max}(f), \qquad s_{\min}(f) \leqslant s_{\max}(f),$$

$$l_{\max}(f) \leqslant n, \quad \text{BPD}(f) \leqslant n \qquad \text{for all } f \in B_n.$$

The following proposition implies $c_{\min}(f) \leqslant s_{\min}(f)$ and $c_{\max}(f) \leqslant s_{\max}(f)$ for all $f \in B_n$.

Proposition 2.2. *$c(f, \boldsymbol{a}) \leqslant s(f, \boldsymbol{a})$ for all $f \in B_n$ and $\boldsymbol{a} \in \{0, 1\}^n$.*

Proof. Let $k = c(f, \boldsymbol{a})$. We show that f is k-sensitive at $\boldsymbol{a}$. Each $(n-k+1)$-dimensional subcube of $\{0, 1\}^n$ containing $\boldsymbol{a}$ contains $n-k+1$ neighbours of $\boldsymbol{a}$ and therefore, at least one of the k neighbours $\boldsymbol{b}$ of $\boldsymbol{a}$ where $f(\boldsymbol{b}) \neq f(\boldsymbol{a})$. □

Proof of Theorem 2.1 (*continued*). *Proof of $l_{\min}(f) = s_{\min}(f)$*: Let $k = s(f, \boldsymbol{a})$. Since f is not $(k+1)$-sensitive at $\boldsymbol{a}$, we find an $(n-k)$-dimensional subcube S (containing

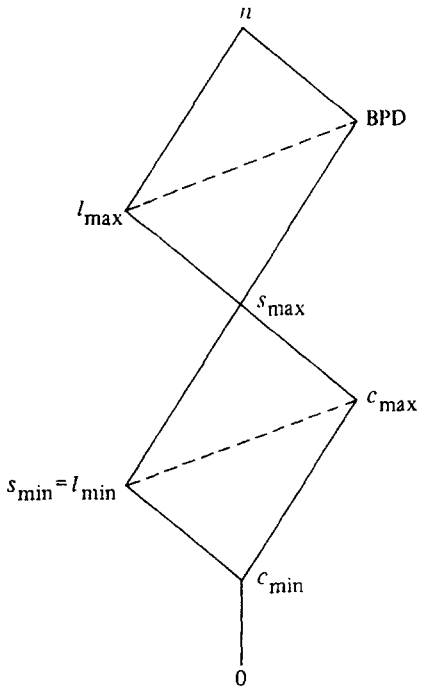

Fig. 1.

a) such that f is constant on S. This subcube corresponds to an implicant (if $f(a)=1$) or a clause (if $f(a)=0$) of f of length k. If, in particular, $s(f, a)=s_{min}(f)$, we have shown that $l_{min}(f) \leq s_{min}(f)$.

On the other hand, let $k=l_{min}(f)$ and let m be, w.l.o.g., a prime implicant of f of length k. Then $S=m^{-1}(1)$ is a subcube of $\{0,1\}^n$ of dimension $n-k$ where f is constant. For $a \in S$ we find an $(n-k)$-dimensional subcube containing a—namely S—where f is constant. Thus, $s_{min}(f) \leq s(f, a) \leq k=l_{min}(f)$.

Proof of $s_{max}(f) \leq l_{max}(f)$: Let $k=s_{max}(f)=s(f, a)$. By our considerations above, we find an implicant or a clause m of length k corresponding to an $(n-k)$-dimensional subcube S containing a where f is constant. If m were not prime, we would obtain an $(n-k+1)$-dimensional subcube $S' \supseteq S$ containing a where f is constant. This contradicts the assumption $s(f, a)=k$. Thus, $s_{max}(f)=k \leq l_{max}(f)$.

Proof of $s_{max}(f) \leq \mathrm{BPD}(f)$: Let $k=s_{max}(f)=s(f, a)$ and let BP be a branching program of minimal depth for the computation of f. We prove the assertion by proving that the computation path of BP for input a has length $l \geq k$. Having reached the leaf for input a in BP we know the value of at most l input bits. Therefore, all inputs of an $(n-l)$-dimensional subcube S containing a reach the same leaf, which implies that f is constant on S. Since $s(f, a)=k$, we conclude that $\mathrm{BPD}(f) \geq l \geq k=s_{max}(f)$.

We add a relation between $l_{max}(f)$ and $c_{max}(f)$ showing that $c_{max}(f)$ cannot be very small if $l_{max}(f)$ is rather large. The most important part of this theorem is reformulated as a corollary.

Theorem 2.3. *For all Boolean functions* $f \in B_n$ $l_{\max}(f) \leq c_{\max}(f)2^{n-l_{\max}(f)}$.

Corollary 2.4. *For all Boolean functions* $f \in B_n$ $l_{\max}(f) = n$ *iff* $s_{\max}(f) = n$ *iff* $c_{\max}(f) = n$.

Corollary 2.4 easily follows from Theorem 2.3 and the already proved parts of Theorem 2.1.

Proof of Theorem 2.3. Let $k = l_{\max}(f)$. Then we find an $(n-k)$-dimensional subcube S where f is constant such that f is not constant on any subcube S' which properly contains S. By definition,

$$\sum_{a \in S} c(f, a) \leq c_{\max}(f) \cdot \# S = c_{\max}(f)2^{n-k}.$$

On the other hand, we have k dimensions to increase S, but in each dimension we find a neighbour b of some $a \in S$ where $f(a) \neq f(b)$. Thus $\sum_{a \in S} c(f, a) \geq k$. Putting the inequalities together we have proved Theorem 2.3. □

Proof of Theorem 2.1 (*continued*). In order to complete the proof of part (a) of Theorem 2.1 we have to show differences between the complexity measures. In doing so we try to maximize the differences. In Fig. 1 we work bottom-up.

(1) $c_{\min}(f) = n$, for the parity function $x_1 \oplus \cdots \oplus x_n$.

(2) $c_{\min}(f) = 0$, but $c_{\max}(f) = n$ for $x_1 \wedge \cdots \wedge x_n$.

(3) $c_{\min}(f) = 0$, but $l_{\min}(f) = s_{\min}(f) = n-1$ for the symmetric function f whose value vector $v(f)$ equals the string 001100... (for a proof of these properties see Section 4).

(4) $s_{\min}(f) = l_{\min}(f) = 1$, but $c_{\max}(f) = s_{\max}(f) = n$ for $x_1 \wedge \cdots \wedge x_n$.

(5) One might think that $s_{\min}(f) = l_{\min}(f) \leq c_{\max}(f)$ for all $f \in B_n$. By systematic search we found the following counterexample where we define the function $f \in B_4$ by its Karnaugh diagram (see Table 1). It can be easily seen that each input a has exactly two neighbours b where $f(b) \neq f(a)$. Thus, $c_{\max}(f) = 2$. The largest subcubes where f is constant have dimension 1. All prime implicants and prime clauses have length 3, i.e., $s_{\min}(f) = l_{\min}(f) = 3 > c_{\max}(f)$. Paterson (personal communication) generalized this example and defined the Boolean function $f \in B_6$ described by

Table 1.

f	00	01	11	10
00	0	1	1	1
01	0	0	0	1
11	1	1	0	1
10	0	1	0	0

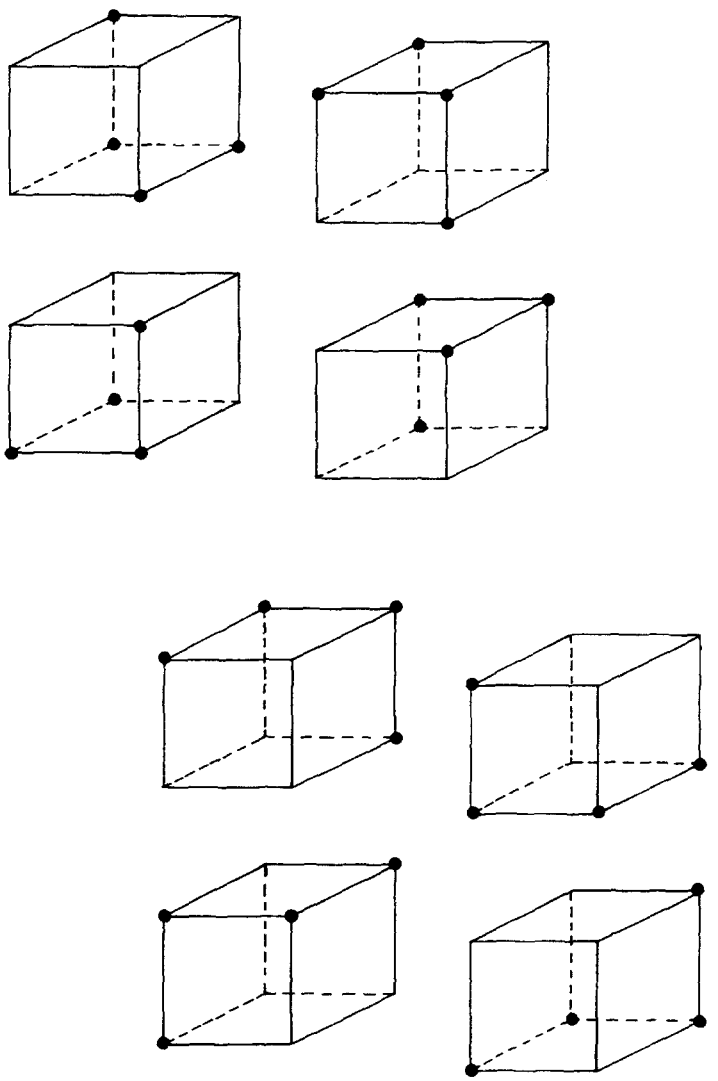

Fig. 2. A Boolean function $f \in B_6$ where $s_{\min}(f) = 5$ and $c_{\max}(f) = 3$.

Fig. 2. It is easy but tedious to prove that $s_{\min}(f) = 5$ and $c_{\max}(f) = 3$. We pump up this example.

W.l.o.g., let $n = 6k$. Then f_k is a Boolean function on n variables. $f_1 \equiv f$ and $f_k \equiv f(X^1) \oplus \cdots \oplus f(X^k)$ where $X^1, \ldots, X^k$ are $k = \frac{1}{6}n$ disjoint blocks of six variables each. We claim $s_{\min}(f_k) = \frac{5}{6}n$ and $c_{\max}(f) = \frac{1}{2}n$. The claim has already been proved for $k = 1$. Let $(\boldsymbol{a}^1, \ldots, \boldsymbol{a}^k) = \boldsymbol{a} \in \{0, 1\}^n$ for $n = 6k$ and let $\boldsymbol{b}$ be a neighbour of $\boldsymbol{a}$ where $f_k(\boldsymbol{a}) \neq f_k(\boldsymbol{b})$. Then, $f(\boldsymbol{a}^i) \neq f(\boldsymbol{b}^i)$ for that i where $d(\boldsymbol{a}^i, \boldsymbol{b}^i) = 1$. Since $c_{\max}(f) = 3$, there are for each block i at most three of those neighbours. Hence, $c_{\max}(f_k) \leq 3k$. If $\boldsymbol{a}^1, \ldots, \boldsymbol{a}^k$ are chosen in such a way that $c(f, \boldsymbol{a}^i) = 3$ for all i, $c(f, \boldsymbol{a}) = 3k$. Hence, $c_{\max}(f) = 3k = \frac{1}{2}n$.

In order to know the value of some $f(\boldsymbol{a}^i)$ we have to know the value of at least five bits of $\boldsymbol{a}^i$, because $s_{\min}(f) = 5$. In order to know $f_k(\boldsymbol{a}) = f(\boldsymbol{a}^1) \oplus \cdots \oplus f(\boldsymbol{a}^k)$, we have to know all values $f(\boldsymbol{a}^i)$, $1 \leq i \leq k$. Hence, $s_{\min}(f_k) \geq 5k$. Since $s_{\min}(f) = 5$, there exists a vector $\boldsymbol{b} \in \{0, 1\}^6$ with $s(f, \boldsymbol{b}) = 5$. Now, choose $\boldsymbol{a}^i = \boldsymbol{b}$, $1 \leq i \leq k$. Then, $s(f_k, \boldsymbol{a}) \leq 5k$. Altogether, $s_{\min}(f_k) = 5k = \frac{5}{6}n$.

We do not know how wide the difference between $s_{\min}$ and $c_{\max}$ may be. Does there exist a sequence $(f_n)_{n\geq 1}$ such that $c_{\max}(f_n) = o(s_{\min}(f_n))$?

(6) $c_{\max}(f) = \lfloor \frac{1}{2}n \rfloor + 2$, but $s_{\max}(f) = n-1$ for the symmetric function f whose value vector equals $(v_0, \ldots, v_n)$ where $v_i = 1$ iff $i \in \{\lfloor \frac{1}{2}n \rfloor, \lfloor \frac{1}{2}n \rfloor + 1\}$ (for a proof see Section 4).

(7) The *address function* $AD_n \in B_n$ for $n = 2^k + k$ is defined in the following way. The variables $y_0, \ldots, y_{m-1}$ for $m = 2^k$ represent the content of m memory cells $M_0, \ldots, M_{m-1}$. The index of the cell is its address. The vector of variables $(x_0, \ldots, x_{k-1})$ is interpreted as the address $|x|$, viz. the integer represented by the binary number $(x_{k-1}, \ldots, x_0)$. Then $AD_n(x, y) = y_{|x|}$ is the content of the memory cell with address $|x|$.

$c_{\max}(f) = s_{\max}(f) = BPD(f) = \lceil \log n \rceil$, but $l_{\max}(f) = n - \lfloor \log n \rfloor$ for $f = AD_n$. The largest prime clause is $y_0 \vee \cdots \vee y_{m-1}$ and the largest prime implicant is $y_0 \wedge \cdots \wedge y_{m-1}$. Thus, $l_{\max}(f) = m = n - k = n - \lfloor \log n \rfloor$. Obviously, $\lceil \log n \rceil = k+1$. $BPD(AD_n) \leq k+1$ since we can compute AD_n by first testing the k address variables $x_0, \ldots, x_{k-1}$ and afterwards asking for the content of $y_{|x|}$.

$c_{\max}(f) \geq k+1$ since the following input is $(k+1)$-critical. Let $|x| = 0$, $y_0 = 0$, $y_1 = \cdots = y_{m-1} = 1$. If we change one of the x-variables or y_0, we obtain an input in AD_n^{-1}. Since $c_{\max}(f) \leq s_{\max}(f) \leq BPD(f)$ for all $f \in B_n$ we have proved our assertions on AD_n. This is the largest known difference since

$$c_{\max}(f) \geq \tfrac{1}{2}\log n - O(\log \log n)$$

for all $f \in B_n$ depending essentially on all their variables [8]. The only function f known that has $c_{\max}(f) < \log n$ is the monotone address function MAD_n (see [12] and Section 3); but for $f = MAD_n$, as for all monotone functions, it holds that $c_{\max}(f) = s_{\max}(f) = l_{\max}(f)$.

(8) $c_{\max}(f) = s_{\max}(f) = l_{\max}(f) = \lceil n^{1/2} \rceil$, but $BPD(f) = n$ for the following function f. We divide the set of n variables into $m = \lceil n^{1/2} \rceil$ disjoint classes $C_1, \ldots, C_m$ each containing at most m variables. f computes 1 iff some class of variables contains only ones. We have m prime implicants each containing the variables of some class C_i and therefore at most m variables. Each prime clause contains exactly one variable of each class and therefore m variables. Thus $l_{\max}(f) = \lceil n^{1/2} \rceil$. Note that $f \in M_n$, i.e., f is monotone. It has been shown in [12] that $c_{\max}(f) = s_{\max}(f) = l_{\max}(f)$ for monotone functions. For each branching program we consider the path p where the first $|C_i| - 1$ variables of C_i tested on this path equal 1 and the last variable equals 0. For this input a, by definition, $f(a) = 0$. If we have tested on p less than n variables, we do not know that f has value 0. Thus the length of p is n.

(9) We have already mentioned that the smallest values of $c_{\max}(f)$, $s_{\max}(f)$, and $l_{\max}(f)$ for functions f depending essentially on n variables are of size $(\frac{1}{2}\log n + O(\log \log n))$. We also have seen that $BPD(AD_n) = \lceil \log n \rceil$. If f depends essentially on all its n variables, $BPD(f) \geq \lceil \log(n+1) \rceil$ since each variable has to be tested somewhere.

Now we have completed the proof of Theorem 2.1(a).

In the rest of this section we shall study the asymptotic behaviour of the above complexity measures, i.e., we shall determine the complexity (for all complexity measures) of almost all Boolean functions. In the following theorem we shall summarize the asymptotic results.

Theorem 2.5. (a) $\mathrm{BPD}(f) = n$ *for almost all Boolean functions* f.

(b) $$\lim_{n\to\infty} (\# B_n)^{-1}(\#\{f \in B_n \mid c_{\max}(f) = n-1\}) = e^{-1};$$

$$\lim_{n\to\infty} (\# B_n)^{-1}(\#\{f \in B_n \mid c_{\max}(f) = n\}) = 1 - e^{-1}.$$

(c) $l_{\max}(f) = s_{\max}(f) = c_{\max}(f)$ *for almost all Boolean functions* f.

(d) *Let* $\alpha(n)$ *be any function tending to* ∞ *as* $n \to \infty$. *Then for almost all Boolean functions*

$$n - \lceil \log(n + \log^2 n - \log n + \alpha(n)) \rceil < l_{\min}(f) = s_{\min}(f) \leqslant n - \lfloor \log(n - \log n - \alpha(n)) \rfloor.$$

Proof of Theorem 2.1 (*conclusion*). From parts (b) and (d) of Theorem 2.5 and example (5) we get Theorem 2.1(b). From parts (a) and (b) of Theorem 2.5 and example (7) we get Theorem 2.1(c). □

Part (a) of Theorem 2.5 has been proved by Rivest and Vuillemin [7] and part (c) follows from part (b), Corollary 2.4, and Theorem 2.1(a). Only for the sake of completeness we mention that $c_{\min}$ has an asymptotic behaviour dual to $c_{\max}$, i.e., the fraction of functions with $c_{\min}(f) = 0$, respectively $c_{\min}(f) = 1$ is asymptotically $1 - e^{-1}$, respectively e^{-1}.

Before proving Theorem 2.5 we shall explain the assertion of part (d). The interval in which $l_{\min}(f) = s_{\min}(f)$ falls for almost all Boolean functions contains at most two integers. For all n not lying near a power of 2, the interval contains the integer $n - \lfloor \log n \rfloor$ only.

We split the proof of Theorem 2.5(b) into the proof of the following two lemmas where the second lemma gives even more information on the number of n-critical inputs.

Lemma 2.6. *For almost all Boolean functions* $c_{\max}(f) \geqslant n - 1$.

Lemma 2.7. *The fraction of functions* $f \in B_n$ *with exactly* k *n-critical inputs converges to* $e^{-1}/(k!)$ (*Poisson distribution with parameter* $\lambda = 1$).

Proof of Lemma 2.6. We make use of a probabilistic argument. Consider 2^n independent $(\frac{1}{2}, \frac{1}{2})$-coin tosses. Each sample of this experiment corresponds to a random

value-table of a Boolean function or, in other words, to a random colouring of the vertices of the n-cube with the colours 0 and 1. Each colouring has the same probability 2^{-2^n}. The following random variables are of interest:

$$X_{a,k}(f)=\begin{cases}1 & \text{if } c(f,a)=k,\\ 0 & \text{otherwise,}\end{cases} \qquad \text{for } a\in\{0,1\}^n f\in B_n,\ 0\leqslant k\leqslant n;$$

$$X_k(f)=\sum_{\substack{a\in\{0,1\}^n,\\ f\in B_n}} X_{a,k}(f) \quad \text{for } f\in B_n,\ 0\leqslant k\leqslant n.$$

$X_k(f)$ counts the number of k-critical points of f. For the proof of Lemma 2.6 it is sufficient to prove that $\lim_{n\to\infty}\Pr(X_{n-1}=0)=0$. By this claim we can conclude

$$\lim_{n\to\infty}\Pr(c_{\max}(f)\geqslant n-1)\geqslant\lim_{n\to\infty}\Pr(X_{n-1}>0)=1.$$

We use the so-called second-moment method. From Chebyshev's inequality (see, e.g., [3]) it follows that

$$\begin{aligned}\Pr(X_{n-1}=0)&\leqslant\Pr(|X_{n-1}-\mathrm{E}(X_{n-1})|\geqslant\mathrm{E}(X_{n-1}))\\ &\leqslant\mathrm{V}(X_{n-1})/\mathrm{E}^2(X_{n-1})=\mathrm{E}(X_{n-1}^2)/\mathrm{E}^2(X_{n-1})-1;\end{aligned}$$

thus it suffices to show that $\mathrm{E}(X_{n-1}^2)/\mathrm{E}^2(X_{n-1})=1+o(1)$.

First we compute the expectation $\mathrm{E}(X_{n-1})$. $\mathrm{E}(X_{a,n-1})=\Pr(X_{a,n-1}=1)$ as $X_{a,n-1}$ is a random variable taking values 0 and 1 only. A point a is $(n-1)$-critical iff exactly $n-1$ neighbours of a have another colour than a itself. There are n possibilities for choosing these neighbours and the probability that after this choice all neighbours of a have the right colour is 2^{-n}. Thus, $\mathrm{E}(X_{a,n-1})=n2^{-n}$ and $\mathrm{E}(X_{n-1})=n$.

Next we compute $\mathrm{E}(X_{n-1}^2)$. Write

$$\begin{aligned}X_{n-1}^2=\sum_a X_{a,n-1}&+\sum_{\substack{a,b,\\ d(a,b)=1}} X_{a,n-1}\cdot X_{b,n-1}+\sum_{\substack{a,b,\\ d(a,b)=2}} X_{a,n-1}\cdot X_{b,n-1}\\ &+\sum_{\substack{a,b,\\ d(a,b)\geqslant 3}} X_{a,n-1}\cdot X_{b,n-1}.\end{aligned}$$

If $d(a,b)=1$, then $\Gamma(a)\cap B(b)$ contains only b. Distinguishing whether a and b have the same colours yields

$$\mathrm{E}(X_{a,n-1}\cdot X_{b,n-1})=\tfrac{1}{2}2^{-2(n-1)}+\tfrac{1}{2}(n-1)^2\cdot 2^{-2(n-1)}=(n^2-2n+2)2^{-2n+1}.$$

If $d(a,b)=2$, the balls $B(a)$ and $B(b)$ have exactly two vertices v, w in common. Distinguishing the possible colour patterns of v and w yields

$$\mathrm{E}(X_{a,n-1}\cdot X_{b,n-1})=4\cdot 2^{-2n+1}+(n-2)^2\cdot 2^{-2n+1}=(n^2-4n+8)2^{-2n+1}.$$

If $d(a,b)\geqslant 3$, the balls $B(a)$ and $B(b)$ are disjoint and hence,

$$\mathrm{E}(X_{a,n-1}\cdot X_{b,n-1})=\mathrm{E}(X_{a,n-1})\cdot\mathrm{E}(X_{b,n-1})=n^2\cdot 2^{-2n}.$$

The number of ordered pairs (a, b) with $d(a, b) = k > 0$ is obviously $\binom{n}{k} \cdot 2^n = O(n^k \cdot 2^n)$. Altogether, $E(X_{n-1}^2) \leq E(X_{n-1}) + O(n^3 \cdot 2^{-n}) + O(n^4 \cdot 2^{-n}) + E^2(X_{n-1})$, i.e.,

$$E(X_{n-1}^2)/E^2(X_{n-1}) = 1 + n^{-1} + O(n^2 \cdot 2^{-n}),$$

which finishes the proof of Lemma 2.6. □

Remark 2.8. For our purposes it was sufficient to investigate X_{n-1}, the number of $(n-1)$-critical vertices. By similar methods we can prove that $\Pr(X_k = 0) = O(\binom{n}{k}^{-1} + n^2 2^{-n})$ for all k. Therefore, almost all Boolean functions simultaneously have k-critical vertices for all $k \in \{1, \ldots, n-1\}$. We shall see later that not almost all Boolean functions have an n-critical vertex (respectively 0-critical vertex).

We want to give a motivation for Lemma 2.7. What is the probability that $c_{\max}(f) = n$? The probability that $c(f, a) = n$ is 2^{-n} and therefore rather small, but we have 2^n possible vectors which may be n-critical. We have seen that $X_{a,n}$ and $X_{b,n}$ are often independent and otherwise 'nearly independent'. If they were really independent, X_n would be asymptotically Poisson-distributed with parameter $\lambda = 1$. The proof of Lemma 2.7 will show that this intuition remains correct despite the small dependencies between the random variables.

Proof of Lemma 2.7. We consider the same probabilistic experiment and the same random variables as in the proof of Lemma 2.6. We shall show that X_n is asymptotically Poisson-distributed with parameter $\lambda = 1$. It is well known that the Poisson distribution is uniquely determined by its factorial moments (for an explicit proof see [2]). Therefore, it is sufficient to prove the following claim for fixed $r \in \mathbb{N}$.

Claim. $\lim_{n \to \infty} E_r(X_n) = 1$ *where* $E_r(X_n) = E(X_n(X_n - 1) \cdot \cdots \cdot (X_n - r + 1))$ *is the r-th factorial moment of* X_n.

By $DIS_{n,r}$ we denote the set of r-tuples $(a^0, \ldots, a^{r-1}) \in (\{0, 1\}^n)^r$ having mutually distinct entries. Then it is easy to prove by induction that

$$E_r(X_n) = E_r\left(\sum_{a \in \{0,1\}^n} X_{a,n}\right) = \sum_{(a^0, \ldots, a^{r-1}) \in DIS_{n,r}} E\left(\prod_{0 \leq j \leq r-1} X_{a^j,n}\right).$$

We split $DIS_{n,r}$ into $INDEP_{n,r} = \{(a^0, \ldots, a^{r-1}) \in DIS_{n,r} \mid d(a^j, a^k) \geq 3 \text{ for } j \neq k\}$ and $DEP_{n,r} = DIS_{n,r} - INDEP_{n,r}$. The balls $B(a^0), \ldots, B(a^{r-1})$ are mutually disjoint if $(a^0, \ldots, a^{r-1}) \in INDEP_{n,r}$ implying, for these $(a^0, \ldots, a^{r-1})$,

$$E\left(\prod_{0 \leq j \leq r-1} X_{a^j,n}\right) = \prod_{0 \leq j \leq r-1} E(X_{a^j,n}) = 2^{-rn}.$$

We now estimate the cardinality of $INDEP_{n,r}$. For $(a^0, \ldots, a^{r-1}) \in INDEP_{n,r}$ we may arbitrarily choose $a^0 \in \{0, 1\}^n$. The choice of $a^0, \ldots, a^{j-1}$ excludes at most

$j(1+n+\binom{n}{2})$ possible choices for a^j. Hence,

$$2^m \geq \# \text{ INDEP}_{n,r} \geq 2^m \prod_{0 \leq j \leq r-1} \left(1 - \frac{j(1+n+\binom{n}{2})}{2^n}\right) = 2^m c_{n,r}$$

where $\lim_{n\to\infty} c_{n,r} = 1$. Combining with the result above, we have

$$\lim_{n\to\infty} \sum_{(a^0,\ldots,a^{r-1}) \in \text{INDEP}_{n,r}} \text{E}\left(\prod_{0 \leq j \leq r-1} X_{a^j,n}\right) = 1. \tag{1}$$

If $(a^0, \ldots, a^{r-1}) \in \text{DEP}_{n,r}$, there exist k and $j < k$ such that $d(a^j, a^k) \leq 2$. There are $\binom{r}{2}$ possibilities for the choice of j and k and less than $2^{n(r-1)}$ possibilities for the choice of all a^m where $m \neq k$. Finally, there are at most $n+\binom{n}{2}$ possibilities for the choice of a^k such that $a^k \neq a^m$ for $k \neq m$ and $d(a^j, a^k) \leq 2$. Thus,

$$\# \text{ DEP}_{n,r} \leq \binom{r}{2} 2^{n(r-1)} \left(n + \binom{n}{2}\right) \leq r^2 n^2 2^{-n} 2^m.$$

We would like to bound $\text{E}(\prod_{0 \leq j \leq r-1} X_{a^j,n})$ for $(a^0, \ldots, a^{r-1}) \in \text{DEP}_{n,r}$. For this purpose we use the well-known fact that the boundary of $(a^0, \ldots, a^{r-1}) \in \text{DIS}_{n,r}$, i.e., the union of all $\Gamma(a^j)$ minus $\{a^0, \ldots, a^{r-1}\}$, is always large even if the distance between each pair of vertices is small. In particular, this follows from the Kruskal-Katona theorem (cf., e.g., [5]).

Fact. *Let $r < n$. The boundary of $(a^0, \ldots, a^{r-1}) \in \text{DIS}_{n,r}$ contains at least $r \cdot n - c_r$ vertices, where $c_r = \binom{r}{2} + r - 1$.*

This yields

$$\text{E}\left(\prod_{0 \leq j \leq r-1} X_{a^j,n}\right) \leq 2^{-(rn-c_r)} \quad \text{for } (a^0, \ldots, a^{r-1}) \in \text{DEP}_{n,r}.$$

Together with our bound on $\# \text{ DEP}_{n,r}$ we obtain

$$\lim_{n\to\infty} \sum_{(a^0,\ldots,a^{r-1}) \in \text{DEP}_{n,r}} \text{E}\left(\prod_{0 \leq j \leq r-1} X_{a^j,n}\right) = 0. \tag{2}$$

Finally, the claim follows from (1) and (2) completing the proof of Lemma 2.7. □

Remark 2.9. In Lemma 2.7 we have assumed that each vertex has probability $\frac{1}{2}$ of being coloured with 0 respectively 1. For the nonuniform distribution with probability p for colour 1 we can prove the following results by similar methods. If $p \neq \frac{1}{2}$ and constant, the probability for the event $c_{\max}(f) = n$ tends to 1. If $p = p(n) = \frac{1}{2} + h(n)$, where $nh(n)$ tends to $r/2$, then X_n is Poisson distributed with parameter $\lambda = \frac{1}{2}(e^r + e^{-r})$. For $r = 0$ (where $h(n) = 0$) we obtain Lemma 2.7.

Proof of Theorem 2.5(d). We consider again the random colouring used for the proofs of Lemma 2.6 and Lemma 2.7. We want to prove that $\Pr(l_{\min}(f) \leq n - c(n))$ tends to 1 for $c(n) = \lfloor \log(n - \log n - \alpha(n)) \rfloor$ and tends to 0 for $c(n) = \lceil \log(n + \log^2 n - \log n + \alpha(n)) \rceil$.

For our first aim we observe that $l_{\min}(f) \leq n - c(n)$ iff there exists a $c(n)$-dimensional subcube of $\{0,1\}^n$ coloured by one colour. We partition $\{0,1\}^n$ into $2^{n-c(n)}$ disjoint $c(n)$-dimensional subcubes C_i $(1 \leq i \leq 2^{n-c(n)})$. Then the events E_i: 'all vertices of C_i have the same colour' are independent. Thus,

$$\Pr(l_{\min}(f) > n - c(n)) \leq \Pr(\forall i: \bar{E}_i) = \prod_{1 \leq i \leq 2^{n-c(n)}} \Pr(\bar{E}_i).$$

Obviously, $\Pr(E_i) = 2^{-2^{c(n)}+1}$. Therefore,

$$\begin{aligned}\Pr(l_{\min}(f) \leq n - c(n)) &\geq 1 - (1 - 2^{-2^{c(n)}})^{2^{n-c(n)}} \\ &= 1 - ((1 - 2^{-2^{c(n)}})^{2^{2^{c(n)}}})^{2^{n-c(n)-2^{c(n)}}}.\end{aligned}$$

The right-hand side of this inequality tends to 1 if $n - c(n) - 2^{c(n)}$ tends to infinity. This is fulfilled for $c(n) = \lfloor \log(n - \log n - \alpha(n)) \rfloor$ and $\alpha(n)$ tending to infinity. This proves our upper bound on $l_{\min}(f) = s_{\min}(f)$.

For our second aim we observe that a function f where $l_{\min}(f) \leq n - c(n)$ has to possess an implicant or a clause of length $n - c(n)$. An implicant or a clause of length $n - c(n)$ determines the value of f on $2^{c(n)}$ vertices of $\{0,1\}^n$. Therefore, there are $2^{2^n - 2^{c(n)}}$ functions $f \in B_n$ with the same fixed implicant or clause of length $n - c(n)$. Furthermore, there are $2\binom{n}{n-c(n)}2^{n-c(n)}$ different implicants and clauses. Thus,

$$\begin{aligned}\Pr(l_{\min}(f) \leq n - c(n)) &\leq 2^{-2^n} 2 \binom{n}{n - c(n)} 2^{n-c(n)} 2^{2^n - 2^{c(n)}} \\ &\leq 2^{n+1+c(n)(\log n - 1) - 2^{c(n)}}.\end{aligned}$$

The right-hand side of this inequality tends to 0 if $2^{c(n)} - (n + 1 + c(n)(\log n - 1))$ tends to infinity. This happens for $c(n) = \lceil \log(n + \log^2 n - \log n + \alpha(n)) \rceil$. Therefore, $\Pr(l_{\min}(f) > n - c(n))$ tends to 1. □

3. Monotone functions

By analogy with Section 2 we shall first consider the relations between the different measures in the case of monotone functions and afterwards present the asymptotic results.

In Section 2 we have proved that $c_{\max}(f) \leq l_{\max}(f)$ for all Boolean functions. As was shown by the address functions AD_n, there can be a large difference between $c_{\max}$ and $l_{\max}$ ($c_{\max}(\mathrm{AD}_n) = \lceil \log n \rceil$, but $l_{\max}(\mathrm{AD}_n) = n - \lfloor \log n \rfloor$). A central observation for monotone functions [12] is that here equality holds, i.e., $c_{\max}(f) = l_{\max}(f)$ for all monotone Boolean functions. This implies the diagram shown in Fig. 3, excepted the proof that complexity measures on different levels of this diagram are

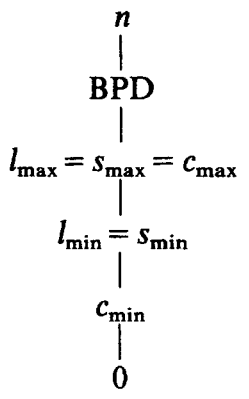

Fig. 3.

indeed different for some monotone functions. This assertion can be proved by a single function.

The *threshold function* T_k^n computes 1 iff the input contains at least k ones. We have $c_{min}(T_2^n) = 0$, $s_{min}(T_2^n) = l_{min}(T_2^n) = 2$, $c_{max}(T_2^n) = s_{max}(T_2^n) = l_{max}(T_2^n) = n-1$, $BPD(T_2^n) = n$. We try again to determine how wide the differences can be. In Fig. 3 we work upside down.

(1) $BPD(f) \leq \log n + \frac{1}{2}\log\log n + O(1) \ll n$ for $f = MAD_n$, the monotone address function. MAD_n is defined for $n = 2k + \binom{2k}{k}$ as

$$MAD_n(x_1, \ldots, x_{2k}, y_1, \ldots, y_{\binom{2k}{k}}) = T_{k+1}^{2k}(x_1, \ldots, x_{2k}) \vee \bigvee_{A \in \mathcal{A}} \bigwedge_{i \in A} (x_i \wedge y_{l(A)}).$$

Here, $\mathcal{A} = \{A \subseteq (1, \ldots, 2k) \mid \#A = k\}$ and l is a one-to-one mapping between $\mathcal{A}$ and $\{1, \ldots, \binom{2k}{k}\}$. MAD_n may be computed by a branching program of depth $2k+1$ in the following way. At first all x-variables are tested. If the number of ones is unequal to k, we know the value of MAD_n. Otherwise, we test $y_{l(A)}$ where $A = \{i \mid x_i = 1\}$ in order to know the value of MAD_n.

(2) $l_{max}(f) = s_{max}(f) = c_{max}(f) = \lceil n^{1/2} \rceil$, but $BPD(f) = n$ for the monotone function $f \in M_n$ from example (8) in Section 2.

(3) $l_{min}(f) = s_{min}(f) = 1$, but $l_{max}(f) = s_{max}(f) = c_{max}(f) = n$ for $f(x) = x_1 \wedge \cdots \wedge x_n$.

(4) $c_{min}(f) = 0$, but $l_{min}(f) = s_{min}(f) = \lfloor \frac{1}{2}(n+1) \rfloor$ for $f = T^n_{\lfloor \frac{1}{2}(n+1) \rfloor}$. The result on $l_{min}(f)$ can be easily checked by the results of Section 4. Furthermore, we shall see in this section that for all monotone functions $l_{min}(f) \leq \lfloor \frac{1}{2}(n+1) \rfloor$.

Concluding this first part we remark that we have a surprisingly easy characterization of s_{min} in the monotone case.

Proposition 3.1. *Let* $\mathbf{0}$ *respectively* $\mathbf{1}$ *be the vector consisting of zeros respectively ones only. For all monotone functions* $s_{min}(f) = \min\{s(f, \mathbf{0}), s(f, \mathbf{1})\}$.

Proof. By definition, the left-hand side cannot be larger than the right-hand side. Therefore, it is sufficient to prove $l_{min}(f) \geq \min\{s(f, \mathbf{0}), s(f, \mathbf{1})\}$. Let m be a prime

implicant of f of minimal length k. This prime implicant corresponds to an $(n-k)$-dimensional subcube S of $\{0,1\}^n$ containing 1 where f is constant. Thus, $s(f,1) \leq k$. The same holds for prime clauses and $s(f,\mathbf{0})$. □

Now we come to the asymptotic results. We summarize our results in the following theorem.

Theorem 3.2. (a) *For almost all monotone Boolean functions* $\mathrm{BPD}(f) = n$.

(b) *For almost all monotone Boolean functions* $l_{\max}(f) = \lceil \frac{1}{2}n \rceil + 1$ *and* $l_{\min}(f) = \lfloor \frac{1}{2}n \rfloor - 1$. (*Remember that for all monotone functions* $l_{\max}(f) = s_{\max}(f) = c_{\max}(f)$ *and* $l_{\min}(f) = s_{\min}(f)$.)

(c) *For all monotone functions but the n projections* $c_{\min}(f) = 0$.

The proof of part (c) is left to the reader. For the proofs of parts (a) and (b) we cite the following result in [6].

Definition 3.3. For $\boldsymbol{a} \in \{0,1\}^n$ the *weight* $w(\boldsymbol{a})$ is the number of ones in $\boldsymbol{a}$. For $0 \leq k \leq n$ we call L_k, the set of all vectors $\boldsymbol{a} \in \{0,1\}^n$ where $w(\boldsymbol{a}) = k$, the *k-th level* of $\{0,1\}^n$.

Definition 3.4. For $s \in \{\lfloor \frac{1}{2}n \rfloor, \lceil \frac{1}{2}n \rceil\}$ we define

$$P_n^s = \{f \in M_n \mid w(\boldsymbol{a}) \geq s+2 \Rightarrow f(\boldsymbol{a}) = 1,\ w(\boldsymbol{a}) \leq s-2 \Rightarrow f(\boldsymbol{a}) = 0,$$
$$\#\{\boldsymbol{a} \in f^{-1}(0) \mid w(\boldsymbol{a}) = s+1\} \leq 2^s,$$
$$\#\{\boldsymbol{a} \in f^{-1}(1) \mid w(\boldsymbol{a}) = s-1\} \leq 2^s\}$$

and $M_n^* = P_n^{n/2}$ if n is even and $M_n^* = P_n^{\lfloor n/2 \rfloor} \cup P_n^{\lceil n/2 \rceil}$ if n is odd.

Theorem 3.5 (Korshunov [6]). *Almost all monotone functions* $f \in M_n$ *are in* M_n^*.

By Korshunov's result we may neglect functions that are not in M_n^* for our asymptotic considerations. Therefore, we shall investigate in the rest of this section M_n^* instead of M_n. .

Proof of Theorem 3.2(a),(b). Here we use additionally the following result in [7]. Let $w_k(f) = \#(f^{-1}(1) \cap L_k)$ and define the polynomial p_f in z by $p_f(z) = \sum_{0 \leq k \leq n} w_k(f) z^k$. Then,

$$(1+z) \nmid p_f(z) \Rightarrow \mathrm{BPD}(f) = n. \tag{3}$$

Our aim is to prove that the fraction of functions $f \in P_n^s$ for which $(1+z) \mid p_f(z)$ tends to 0. We define an equivalence relation R on P_n^s.

$$fRf' \Leftrightarrow f(\boldsymbol{a}) = f'(\boldsymbol{a}) \quad \text{for all } \boldsymbol{a} \notin L_s.$$

Let $P_n^s(\alpha)$ be an equivalence class with respect to R. For $f, f' \in P_n^s(\alpha)$ we conclude that $w_k(f) = w_k(f')$ for all $k \neq s$. Since $(1+z) \nmid z^s$, there exists at most one value w^* for $w_s(f)$ such that $(1+z) \mid p_f(z)$ for $f \in P_n^s(\alpha)$.

Because of the monotonicity of the considered functions we cannot arbitrarily define $f \in P_n^s(\alpha)$ on level L_s. But, by the structure of P_n^s, at most $O(n2^{n/2})$ inputs $\boldsymbol{a} \in L_s$ are forced to lie in $f^{-1}(0)$ and at most $O(n2^{n/2})$ inputs $\boldsymbol{a} \in L_s$ are forced to lie in $f^{-1}(1)$. Thus the value of at least $a_n(\alpha, s) = \binom{n}{s} - O(n2^{n/2}) = \Omega(n^{-1/2}2^n)$ inputs $\boldsymbol{a} \in L_s$ can be chosen arbitrarily. At most $\binom{a_n(\alpha,s)}{\lfloor \frac{1}{2}a_n(\alpha,s) \rfloor}$ of the $2^{a_n(\alpha,s)}$ functions in $P_n^s(\alpha)$ have the right value for $w_s(f)$ such that $(1+z) \mid p_f(z)$. The fraction of these functions is bounded by $O(a_n(\alpha, s)^{-1/2}) = O(n^{1/4}2^{-n/2})$ and therefore, uniformly bounded for all classes $P_n^s(\alpha)$. Thus, the fraction of functions $f \in P_n^s$ such that $(1+z) \mid p_f(z)$ is bounded by $O(n^{1/4}2^{-n/2})$ and tends to 0. Finally, by (3) this implies that $\mathrm{BPD}(f) = n$ for almost all monotone functions. In the terminology of [7] this result reads as follows: almost all monotone functions are exhaustive.

Proof of Theorem 3.2(b): We first consider the case that n *is even.* By Theorem 3.5 we may restrict our attention to the class P_n^s where $s = \frac{1}{2}n$. For these functions $f(\boldsymbol{a}) = 0$ if $w(\boldsymbol{a}) < s-1$ and $f(\boldsymbol{a}) = 1$ if $w(\boldsymbol{a}) > s+1$. Since prime implicants correspond to minimal ones and prime clauses correspond to maximal zeros we know that the length of any prime implicant or prime clause may take the values $s-1$, s, $s+1$, and $s+2$ only.

If $l_{\min}(f) \geq s$, we additionally know that $f(\boldsymbol{a}) = 0$ for $w(\boldsymbol{a}) = s-1$ and $f(\boldsymbol{a}) = 1$ for $w(\boldsymbol{a}) = s+1$. Therefore, $l_{\min}(f) \geq s$ for exactly $2^{b(n,s)}$ functions in P_n^s where $b(n, s) = \binom{n}{s}$. Again, by results in [6] on the number of monotone functions, this is a fraction of M_n tending to 0. This implies the result on $l_{\min}(f)$.

Let us count the number of $f \in P_n^s$ such that $l_{\max}(f) \leq s$. We fix the value of f on all $\boldsymbol{a} \in L_s$. Furthermore, $f(\boldsymbol{a}) = 0$ if $w(\boldsymbol{a}) < s-1$ and $f(\boldsymbol{a}) = 1$ if $w(\boldsymbol{a}) > s+1$. By the values of f on L_s we know, by monotonicity, some inputs $\boldsymbol{a} \in L_{s+1}$ where $f(\boldsymbol{a})$ is forced to be 1 and some inputs $\boldsymbol{a} \in L_{s-1}$ where $f(\boldsymbol{a})$ is forced to be 0. The other inputs on L_{s-1} and L_{s+1} are called 'holes' since we may still choose $f(\boldsymbol{a})$ arbitrarily. These holes are independent in the following sense. If we fix $f(\boldsymbol{a}) = 0$ for some hole $\boldsymbol{a} \in L_{s-1}$ this obviously has no influence on other holes. If we fix $f(\boldsymbol{a}) = 1$ for some hole $\boldsymbol{a} \in L_{s-1}$, then, for all neighbours $\boldsymbol{b} \in L_s$, $f(\boldsymbol{b}) = 1$. Otherwise, $\boldsymbol{a}$ would not have been a hole since $f(\boldsymbol{a})$ would have been forced to equal 0. Furthermore, no $\boldsymbol{c} \in L_{s+1}$ where $d(\boldsymbol{a}, \boldsymbol{c}) = 2$ is a hole. Any such $f(\boldsymbol{c})$ is forced to equal 1 by some neighbour $\boldsymbol{b} \in L_s$.

Thus we may choose the value of the holes in an arbitrary way. But it is easy to see that $f(\boldsymbol{a}) = 1$ for some hole $\boldsymbol{a} \in L_{s+1}$ implies that $\boldsymbol{a}$ is a minimal element of $f^{-1}(1)$. Hence, $\boldsymbol{a}$ corresponds to a prime implicant of length $s+1$. Dual arguments hold for holes on level $s-1$.

Altogether, the number of $f \in P_n^s$ such that $l_{\max}(f) \leq s$ is bounded by $2^{b(n,s)}$ and therefore a fraction tending to 0.

Finally, we count the number of $f \in P_n^s$ such that $l_{\max}(f) = s+2$. Such a function has, w.l.o.g., a prime implicant m of length $s+2$ (dual arguments hold for prime

clauses). Let $\boldsymbol{a}$ be the vector in L_{s+2} corresponding to m. Then g defined by $g(\boldsymbol{a})=0$ and $g(\boldsymbol{b})=f(\boldsymbol{b})$ for all $\boldsymbol{b}\neq\boldsymbol{a}$ is a monotone function $g\in M_n-P_n^s$. Since different f, $f'\in P_n^s$ where $l_{\max}(f)=l_{\max}(f')=s+2$ differ already on the levels L_{s-1}, L_s, L_{s+1}, we obtain different g, $g'\in M_n-P_n^s$. Thus the number of $f\in P_n^s$ and $l_{\max}(f)=s+2$ is not larger than the number of $g\in M_n-P_n^s$ and again we have a fraction of all monotone functions tending to 0.

If n *is odd*, we need other arguments. Let Q_n^s for $s=\lfloor\frac{1}{2}n\rfloor$ be the class of all functions $f\in P_n^s$ such that $w(\boldsymbol{a})=s-1$ implies $f(\boldsymbol{a})=0$.

Fact. *Almost all monotone functions are not in* Q_n^s.

Proof. For each mapping $\alpha: L_{s+1}\to\{0,1\}$ where $\#\,\alpha^{-1}(0)\leqslant 2^{n/2}$ we denote by $P_n^s(\alpha)$ respectively $Q_n^s(\alpha)$ the class of all $f\in P_n^s$ respectively Q_n^s such that $f(\boldsymbol{a})=\alpha(\boldsymbol{a})$ for all $\boldsymbol{a}\in L_{s+1}$. We prove the assertion by proving that, for all α, $\#\,Q_n^s(\alpha)/\#\,P_n^s(\alpha)$ is uniformly bounded by some sequence $q_n\to 0$.

For the considered mapping α we have $O(2^{n/2})$ zeros at level $s+1$ implying $O(n2^{n/2})$ zeros on level s and $O(n^2 2^{n/2})$ zeros on level $s-1$. Let $a_n(\alpha)$ be the number of inputs $\boldsymbol{a}\in L_s$ which are not forced to be mapped on 0 by α. Obviously, $\#\,Q_n^s(\alpha)=2^{a_n(\alpha)}$.

Having fixed f on L_s we obtain some holes on level $s-1$ where the value of f may be chosen in an arbitrary way with the only restriction that at most $2^{n/2}$ inputs may have value 1. Let β be any choice for the values of f on L_s consistent with α and let $b_n(\alpha,\beta)$ be the number of holes on L_{s-1}. Then, $\#\,P_n^s(\alpha)\geqslant\sum_\beta b_n(\alpha,\beta)$. Furthermore,

$$\#\,P_n^s(\alpha)/\#\,Q_n^s(\alpha)\geqslant 2^{-a_n(\alpha)}\sum_\beta b_n(\alpha,\beta)=:h(\alpha),$$

where $h(\alpha)$ is the expected number of holes on level $s-1$ for the following experiment. The probability that $f(\boldsymbol{a})=0$ respectively $f(\boldsymbol{a})=1$ equals $\frac{1}{2}$ for all $\boldsymbol{a}\in L_s$ where $f(\boldsymbol{a})$ is not forced by α to equal 0. The inputs are treated independently. For each of the $\binom{n}{s-1}-O(n^2 2^{n/2})=\Omega(n^{-1/2}2^n)$ inputs $\boldsymbol{a}\in L_{s-1}$, where $f(\boldsymbol{a})$ is not forced by α to equal 0, the probability that $\boldsymbol{a}$ becomes a hole equals $2^{-(n-s+1)}$. Therefore,

$$h(\alpha)=2^{-(n-s+1)}\Omega(n^{-1/2}2^n)=\Omega(n^{-1/2}2^{n/2})$$

is uniformly bounded for all α and finally, $\#\,Q_n^s(\alpha)/\#\,P_n^s(\alpha)$ converges uniformly to 0. □

We restrict our attention to P_n^s for $s=\lfloor\frac{1}{2}n\rfloor$ since dual arguments hold for P_n^{s+1}. For functions in P_n^s we know that $f(\boldsymbol{a})=0$ if $w(\boldsymbol{a})\leqslant s-2$, $f(\boldsymbol{a})=1$ if $w(\boldsymbol{a})\geqslant s+2$, $f(\boldsymbol{a})=1$ for at most $2^{n/2}$ inputs $\boldsymbol{a}$ where $w(\boldsymbol{a})=s-1$, and $f(\boldsymbol{a})=0$ for at most $2^{n/2}$ inputs $\boldsymbol{a}$ where $w(\boldsymbol{a})=s+1$. Thus prime implicants have length $l\in\{s-1,s,s+1,s+2\}$ and prime clauses have length $l\in\{s,s+1,s+2,s+3\}$.

If $l_{\min}(f)\geqslant s+1$, we know that $f(\boldsymbol{a})=0$ if $w(\boldsymbol{a})\leqslant s$ and $f(\boldsymbol{a})=1$ if $w(\boldsymbol{a})\geqslant s+1$. The only function with $l_{\min}(f)\geqslant s+1$ is T_{s+1}^n. If $l_{\min}(f)=s$, we know that $f\in Q_n^s$. By the fact above, the fraction of all f where $l_{\min}(f)\geqslant s$ tends to 0.

Finally, we prove our claim for l_{max}. If $l_{max}(f) = s+3$, we have a prime clause of length $s+3$, that means a maximal zero on level $s-2$. Changing f for this input from 0 to 1 we obtain a function $f' \notin P_n^s \cup P_n^{s+1}$. For different functions we obtain different functions. Thus the fraction of functions f where $l_{max}(f) = s+3$ tends to 0.

If $l_{max}(f) = s$, all maximal zeros are lying on level $s+1$, but there are at most $2^{n/2}$ zeros on this level and therefore only $O(n^3 2^{n/2})$ zeros on level $s-2$, but all $\binom{n}{s-2}$ inputs should be mapped to 0. Therefore, these functions f are not in $P_n^s \cup P_n^{s+1}$.

We now investigate the functions $f \in P_n^s$ where $l_{max}(f) = s+1$. We fix the values on level s and $s+1$. Some $f(\boldsymbol{a})$ for $\boldsymbol{a}$ on level $s-1$ are forced to be 0, the other $\boldsymbol{a} \in L_{s-1}$ are called 'holes'. In order to obtain $f \in P_n^s$ we have to fill the holes with ones, otherwise we would obtain too long prime clauses. Thus the number of functions where $l_{max}(f) = s+1$ is equal to the number of functions in Q_n^s and therefore sufficiently small. □

4. Symmetric functions

Recall that a symmetric function $f \in S_n$ can be represented by its value vector $v(f) = (v_0, \ldots, v_n)$ where v_i is the value of f on all inputs with exactly i ones. Sometimes we consider $v(f)$ also as a 0-1-string. It is possible to describe the complexity of f (for all our complexity measures) by properties of the value vector. From these descriptions it is obvious that the complexity of f can be computed in time $O(n)$. Also the relations between the complexity measures and their asymptotic behaviour follow from these descriptions.

It has already been proved [11] that $BPD(f) = n$ for all nonconstant $f \in S_n$. Let $c(k)$ be the critical complexity of inputs with exactly k ones. Then (see [12]),

$$c(k) = \begin{cases} 0 & \text{if } v_{k-1} = v_k = v_{k+1}, \\ k & \text{if } v_{k-1} \neq v_k = v_{k+1}, \\ n-k & \text{if } v_{k-1} = v_k \neq v_{k+1}, \\ n & \text{if } v_{k-1} \neq v_k \neq v_{k+1}, \end{cases} \qquad \text{for } 0 < k < n.$$

If $k = 0$ ($k = n$) we only have to delete v_{k-1} (v_{k+1}).

Theorem 4.1. (a) $BPD(f) = n$ *for all nonconstant* f.

(b) $l_{max}(f) = s_{max}(f) = n+1-v_{min}(f)$ *where* $v_{min}(f)$ *is the length of a shortest maximal constant substring of* $v(f)$.

(c) $c_{max}(f) = n$ *iff* $v(f)$ *starts or ends with* 01 *or* 10 *or contains* 010 *or* 101 *as a substring. Otherwise,*

$$c_{max}(f) = \max\{\max\{k \mid v_{k-1} \neq v_k\}, \max\{n-k \mid v_k \neq v_{k+1}\}, 0\}.$$

(d) $l_{min}(f) = s_{min}(f) = n+1-v_{max}(f)$ *where* $v_{max}(f)$ *is the length of a longest constant substring of* $v(f)$.

(e) $c_{\min}(f)=0$ *iff* $v(f)$ *starts or ends with* 00 *or* 11 *or contains* 000 *or* 111 *as a substring. Otherwise*

$$c_{\min}(f)=\min\{\min\{k \mid v_k=v_{k+1}\},\ \min\{n-k \mid v_{k-1}=v_k\},\ n\}.$$

Proof. We only have to prove parts (b) and (d).

A prime implicant t of length k with l variables and $k-l$ negated variables implies $v_l=\cdots=v_{n-k+l}=1$ and therefore the existence of a constant substring of $v(f)$ of length $n+1-k$. Furthermore, we obtain a maximal constant substring. If $v_{l-1}=1$, respectively $v_{n-k+l+1}=1$, we could shorten t by a variable respectively negated variable. If, on the other hand, $v_l=\cdots=v_{n-k+l}=1$ is a maximal constant substring of $v(f)$, the monom $x_1\ldots x_l\,\bar{x}_{l+1}\ldots\bar{x}_k$ is a prime implicant of f of length k. Dual arguments hold for prime clauses and substrings of $v(f)$ consisting of zeros. Thus, $l_{\max}(f)+v_{\min}(f)=l_{\min}(f)+v_{\max}(f)=n+1$.

Since we already know that $s_{\min}(f)=l_{\min}(f)$, we have proved part (d). We also know that $s_{\max}(f)\leqslant l_{\max}(f)$. Thus, for part (b) it is sufficient to prove that $s_{\max}(f)\geqslant n+1-v_{\min}(f)$.

Let $v_{\min}(f)=k+1$ and $(v_l,\ldots,v_{l+k})$ be a maximal constant substring of $v(f)$. Let $\boldsymbol{a}$ be an input vector with exactly l ones. We claim that $s_{\max}(f)\geqslant s(f,\boldsymbol{a})\geqslant n-k=n+1-v_{\min}(f)$. Let W be a $(k+1)$-dimensional subcube of $\{0,1\}^n$ containing $\boldsymbol{a}$. If f were constant on W, we could conclude that there exists a constant substring $(v_m,\ldots,v_{m+k+1})$ of $v(f)$ of length $k+2$ including v_l. This is a contradiction since $(v_l,\ldots,v_{l+k})$ is a maximal constant substring of $v(f)$. □

We now present the largest possible differences between the complexity measures.

(1) $\mathrm{BPD}(f)=n$, but $l_{\max}(f)=s_{\max}(f)=\lceil\frac{1}{2}(n+1)\rceil$ for $v(f)=0^{\lfloor\frac{1}{2}(n+1)\rfloor}1^{\lceil\frac{1}{2}(n+1)\rceil}$.

(2) $l_{\max}(f)=s_{\max}(f)=n-1$, but $c_{\max}(f)=\lfloor\frac{1}{2}n\rfloor+2$ for $v(f)=(v_0,\ldots,v_n)$ where $v_i=1$ iff $i\in\{\lfloor\frac{1}{2}n\rfloor,\ \lfloor\frac{1}{2}n\rfloor+1\}$.

(3) $c_{\max}(f)=n$, but $s_{\min}(f)=l_{\min}(f)=1$ for $f(x)=x_1\wedge\cdots\wedge x_n$.

(4) $s_{\min}(f)=l_{\min}(f)=n-1$, but $c_{\min}(f)=0$ for $v(f)=001100\ldots$.

(5) $c_{\min}(f)=n$ for $f(x)=x_1\oplus\cdots\oplus x_n$.

By these examples the diagram in Fig. 4 for symmetric functions is complete.

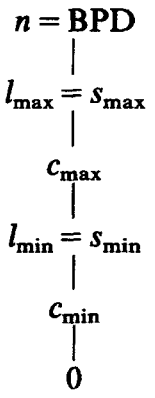

Fig. 4.

At the end of this section we describe the asymptotic behaviour of the complexity measures for symmetric functions.

Theorem 4.2. (a) *For all nonconstant symmetric functions* $f \in S_n$ *we have* $\mathrm{BPD}(f) = n$.

(b) *The number of* $f \in S_n$ *such that* $c_{\max}(f) < n$ *equals*

$$A_n = \frac{2}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n\left(1-\left(\frac{1-\sqrt{5}}{1+\sqrt{5}}\right)^n\right).$$

The same holds for the number of $f \in S_n$ *such that* $s_{\max}(f) = l_{\max}(f) < n$. *For almost all symmetric functions* $l_{\max}(f) = s_{\max}(f) = c_{\max}(f) = n$.

(c) *Let* $\alpha(n)$ *be any function tending to* ∞ *as* $n \to \infty$. *Then, for almost all symmetric functions,* $n - \log n - \alpha(n) \leq l_{\min}(f) = s_{\min}(f) \leq n - \log n + \alpha(n)$.

(d) *For almost all symmetric functions* $c_{\min}(f) = 0$.

Part (a) has already been mentioned before. The result on $c_{\min}$ (part (d)) is left to the reader. By Theorem 4.1, part (c) is equivalent to the assertion that for almost all 0-1-vectors of length $n+1$ the estimate $\log n - \alpha(n) \leq v_{\max}(f) \leq \log n + \alpha(n)$ holds. This assertion follows from well-known investigations on random 0-1-sequences (see, e.g., [3]). By these results one can also compute very exactly the probability that $v_{\max}(f)$ lies in some interval I.

Proof of Theorem 4.2(b). By Corollary 2.4 we know that either $c_{\max}(f) = l_{\max}(f) = s_{\max}(f) = n$ or all these numbers are less than n. Since $\# S_n = 2^{n+1}$, obviously, $A_n / \# S_n$ tends to 0. Therefore, we only have to count those $f \in S_n$ for which $l_{\max}(f) < n$. By Theorem 4.1, $l_{\max}(f) < n$ iff $v_{\min}(f) > 1$. Thus, $l_{\max}(f) < n$ iff $v(f) = 0^{k(1)}1^{k(2)}0^{k(3)} \ldots a^{k(t)}$ or $v(f) = 1^{k(1)}0^{k(2)}1^{k(3)} \ldots (1-a)^{k(t)}$, where all $k(i) > 1$, $a = 0$ if t is odd and $a = 1$ if t is even. Obviously, $A_1 = A_2 = 2$ and $A_{n+1} = A_n + A_{n-1}$. In other words $A_n = 2F_{n-1}$, where F_k are the Fibonacci numbers. The value of the Fibonacci numbers is well known (see, e.g., [4]). □

Acknowledgment

We thank Mike Paterson for helpful comments, in particular for communicating the generalized example (5) in Section 2.

References

[1] S. Cook, C. Dwork and R. Reischuk, Upper and lower time bounds for parallel random access machines without simultaneous writes, *SIAM J. Comput.* **15** (1986) 87-97.

[2] P. Erdös and A. Rényi, On the evolution of random graphs, *Publ. Math. Inst. Hung. Acad. Sci.* **5A** (1960) 17-61.

[3] W. Feller, *An Introduction to Probability Theory and its Applications* (Wiley & Sons, New York, 1968).

[4] R. Kemp, *Fundamentals of the Average Case Analysis of Particular Algorithms* (Wiley/Teubner, Stuttgart, 1984).

[5] D.J. Kleitmann, Extremal hypergraph problems, in: B. Bollobás, ed., *Surveys in Combinatorics* (Cambridge University Press, 1979) 44–65.

[6] A.D. Korshunov, On the number of monotone Boolean functions, *Probl. Kibern.* **38** (1981) 5–108 (in Russian).

[7] R.L. Rivest and J. Vuillemin, On recognizing graph properties from adjacency matrices, *Theoret. Comput. Sci.* **3** (1976) 371–384.

[8] H.U. Simon, A tight $\Omega(\log \log n)$ bound on the time for parallel RAMs to compute nondegenerate Boolean functions, *Proc. FCT '83*, Lecture Notes in Computer Science **158** (1983) 439–444.

[9] G. Turán, The critical complexity of graph properties, *Inform. Process. Lett.* **18** (1984) 151–153.

[10] U. Vishkin and A. Widgerson, Trade-offs between depth and width in parallel computation, *SIAM J. Comput.* **14** (1985) 303–314.

[11] I. Wegener, Optimal decision trees and one-time-only branching programs for symmetric Boolean functions, *Inform. and Control* **62** (1984) 129–143.

[12] I. Wegener, The critical complexity of all (monotone) Boolean functions and monotone graph properties, *Inform. and Control* **67** (1985) 212–222.